

FRIST - Flipping and Rotation Invariant Sparsifying Transform Learning and Applications to Inverse Problems

Bihan Wen¹, Saiprasad Ravishankar², and Yoram Bresler¹

¹ Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, Champaign, IL 61801, USA.

² Department of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109, USA.

E-mail: bwen3@illinois.edu, ravisha@umich.edu, and ybresler@illinois.edu

January 2016

Abstract. Features based on sparse representation, especially using the synthesis dictionary model, have been heavily exploited in signal processing and computer vision. However, synthesis dictionary learning typically involves NP-hard sparse coding and expensive learning steps. Recently, sparsifying transform learning received interest for its cheap computation and its optimal updates in the alternating algorithms. In this work, we develop a methodology for learning of Flipping and Rotation Invariant Sparsifying Transforms, dubbed FRIST, to better represent natural images that contain textures with various geometrical directions. The proposed alternating learning algorithm involves efficient optimal updates. We provide a convergence guarantee, and demonstrate the empirical convergence behavior of the proposed FRIST learning algorithm. Preliminary experiments show the usefulness of adaptive sparse representation by FRIST for image sparse representation, segmentation, denoising, robust inpainting, and MRI reconstruction with promising performances.

Keywords: Sparsifying transform learning, Dictionary learning, Convergence guarantee, Overcomplete representation, Machine learning, Clustering, Image representation, Inverse problem, Image denoising, Inpainting, Magnetic resonance imaging.

1. Introduction

Sparse representation of natural signals in a certain transform domain or dictionary has been widely exploited. Various sparse models, such as the synthesis model [4, 10] and the transform model [25], have been studied. The popular *synthesis model* suggests that a signal $y \in \mathbb{R}^n$ can be sparsely represented as $y = Dx + \eta$, where $D \in \mathbb{R}^{n \times m}$ is a synthesis dictionary, $x \in \mathbb{R}^m$ is a sparse code, and η is small approximation error in the signal domain. Synthesis dictionary learning methods [11, 3] typically involve a synthesis

sparse coding step which is, however, NP-hard [8], so that approximate solutions [21, 19, 6] are widely used. Various dictionary learning algorithms [11, 44, 37, 17], especially the well-known KSVD method [3], have been proposed and are popular in numerous applications such as denoising, inpainting, deblurring, and demosaicing [9, 16, 2]. However, they are typically computationally expensive for large-scale problems. Moreover, heuristic methods such as KSVD can get easily caught in local minima, or saddle points [36].

The alternative *transform model* suggests that the signal y is approximately sparsifiable using a transform $W \in \mathbb{R}^{m \times n}$, i.e., $Wy = x + e$, with $x \in \mathbb{R}^m$ sparse and e a small approximation error in the transform domain (rather than in the signal domain). It is well-known that natural images are sparsifiable by analytical transforms such as the discrete cosine transform (DCT), or wavelet transform [18]. Furthermore, recent works proposed learning square sparsifying transforms (SST) [31], which turn out to be advantageous in various applications such as image denoising, magnetic resonance imaging (MRI), and computed tomography (CT) [31, 32, 23, 24]. Alternating minimization algorithms for learning SST have been proposed with cheap and closed-form updates [35].

Since SST learning is restricted to one adaptive square transform for all data, the diverse patches of natural images may not be sufficiently sparsified in the SST model. Recent work focused on learning a union of unstructured sparsifying transforms [41, 40], dubbed OCTOBOS, to sparsify images with different contents and diverse features. However, the unstructured OCTOBOS may suffer from overfitting in various applications. Instead, we consider in this paper the use of transformation symmetries to constrain the multiple learned transforms, thus reducing the number of free parameters and avoiding overfitting. While previous works exploited transformation symmetries in synthesis model sparse coding [42], and applied rotational operators with analytical transforms [48], the usefulness of the rotational invariance property in learning adaptive sparse models has not been explored. Hence, in this work, we propose a Flipping and Rotation Invariant Sparsifying Transform (FRIST) learning scheme, and show that it can provide better sparse representation by capturing the “optimal” orientations of patches in natural images. As such, it can serve as an effective regularizer for image recovery in various inverse problems. Preliminary experiments in this paper show the usefulness of adaptive sparse representation by FRIST for image sparse representation, segmentation, denoising, robust inpainting, and MRI reconstruction with promising performances.

2. FRIST Model and Its Learning Formulation

FRIST Model. The learning of the sparsifying transform model [31] has been proposed recently. Extending this approach, we propose a *FRIST model* that first applies a flipping and rotation (FR) operator $\Phi \in \mathbb{R}^{n \times n}$ to a signal $y \in \mathbb{R}^n$, and models Φy as approximately sparsifiable by some sparsifying transform $W \in \mathbb{R}^{m \times n}$ as $W\Phi y = x + e$,

with $x \in \mathbb{R}^m$ sparse, and e small. A finite set of flipping and rotation operators $\{\Phi_k\}_{k=1}^K$ is considered, and the sparse coding problem in the FRIST model is as follows,

$$(P1) \quad \min_{1 \leq k \leq K} \min_{x^k} \|W \Phi_k y - x^k\|_2^2 \quad s.t. \quad \|x^k\|_0 \leq s \quad \forall k,$$

where the ℓ_0 “norm” counts the number of nonzeros in x^k . Thus, x^k denotes the sparse code of $\Phi_k y$ in the transform W domain, with maximum sparsity s . Equivalently, the optimal \hat{x}^k is called the sparse code in the FRIST domain. We further decompose the FR matrix as $\Phi_k \triangleq G_q F$, where F can be either an identity matrix, or (for 2D signals) a left-to-right flipping permutation matrix. Though there are various methods of formulating the rotation operator G with arbitrary angles [14, 13], rotating image patches by an angle θ that is not a multiple of 90° requires interpolation, and may result in misalignment with the pixel grid. Here, we adopt the matrix $G_q \triangleq G(\theta_q)$ that permutes the pixels in an image patch approximating rotation by angles $\{\theta_q\}$ without interpolation. Constructions of such $\{G_q\}$ have been proposed before [22, 26, 48]. With such implementation, the number of possible permutations $\{G_q\}$ is \tilde{Q} , which is finite and grows linearly with the data dimension n . Accounting for the flipping operation, the total number of possible FR operators is $\tilde{K} = 2\tilde{Q}$.

In practice, one can select a subset $\{\Phi_k\}_{k=1}^{\tilde{K}}$, containing a constant number $K < \tilde{K}$ of FR candidates, from which the optimal $\hat{\Phi} = \Phi_{\hat{k}}$ is chosen. For each Φ_k , the optimal sparse code \hat{x}^k in Problem (P1) can be solved exactly as $\hat{x}^k = H_s(W\Phi_k y)$, where $H_s(\cdot)$ is the projector onto the s - ℓ_0 ball [29], i.e., $H_s(b)$ zeros out all but the s elements of largest magnitude in $b \in \mathbb{R}^m$. The optimal FR operator $\hat{\Phi}_k$ is selected to provide the smallest sparsification (modeling) error $\|W \Phi_{\hat{k}} y - H_s(W\Phi_{\hat{k}} y)\|_2^2$ in Problem (P1).

The FRIST model can be interpreted as a structured union-of-transforms model, or a structured OCTOBOS model [41]. In particular, we can construct structured sub-transform W_k as $W_k = W\Phi_k$, in which case the collection $\{W_k\}_{k=1}^K$ represents a union-of-transforms model. Thus, each signal y is best sparsified by one particular W_k from the union of transforms $\{W_k\}$. The transforms in the collection all share a common transform W . We call the shared common transform W the **parent transform**, and each generated W_k is called a **child transform**. Problem (P1) is similar to the OCTOBOS sparse coding problem [41], where each W_k corresponds to a block of OCTOBOS. Similar to the clustering procedure in OCTOBOS, Problem (P1) matches a signal y to a particular child transform W_k with its directional FR operator Φ_k . Thus, FRIST is potentially capable of automatically clustering a collection of signals (e.g., image patches), but according to their geometric orientations. When the parent transform W is unitary, FRIST is also equivalent to an overcomplete synthesis dictionary with block sparsity [47], with W_k^T denoting the k th block of the equivalent overcomplete dictionary. Compared to a general overcomplete dictionary or OCTOBOS, FRIST is much more constrained, with fewer free parameters. This property turns out to be useful in inverse problems such as denoising and inpainting, preventing the overfitting of the model in the presence of limited or highly corrupted data or measurements.

FRIST Learning Formulation. Generally, the parent transform W can be overcomplete [30, 41, 24]. In this work, we restrict ourselves to learning FRIST with a square parent transform W (i.e., $m = n$), which leads to a highly efficient learning algorithm with optimal updates. Note that the FRIST model is still overcomplete, even with a square parent W . Given the training data $Y \in \mathbb{R}^{n \times N}$, we formulate the FRIST learning problem as follows

$$\begin{aligned} (\text{P2}) \quad & \min_{W, \{X_i\}, \{C_k\}} \sum_{k=1}^K \left\{ \sum_{i \in C_k} \|W\Phi_k Y_i - X_i\|_2^2 \right\} + \lambda Q(W) \\ & \text{s.t. } \|X_i\|_0 \leq s \quad \forall i, \quad \{C_k\} \in \Gamma \end{aligned}$$

where $\{X_i\}$ represent the FRIST-domain sparse codes of the corresponding columns $\{Y_i\}$ of Y , and $X \in \mathbb{R}^{n \times N}$ with columns X_i denotes the sparse code matrix of Y . The $\{C_k\}$ indicate a clustering of the signals $\{Y_i\}$ such that each signal is associated exactly with one FR operator Φ_k . The set Γ is the set of all possible partitions (into K subsets) of the set of integers $\{1, 2, \dots, N\}$, which enforces all of the C_k 's to be disjoint [41].

Problem (P2) is to minimize the FRIST learning objective that includes the modeling error $\sum_{k=1}^K \left\{ \sum_{i \in C_k} \|W\Phi_k Y_i - X_i\|_2^2 \right\}$ for Y , as well as the regularizer $Q(W) = -\log |\det W| + \|W\|_F^2$ to prevent trivial solutions [31]. Here, the log determinant penalty $-\log |\det W|$ enforces full rank on W , and the $\|W\|_F^2$ penalty helps remove a ‘scale ambiguity’ in the solution. Regularizer $Q(W)$ fully controls the condition number and scaling of the learned parent transform [31]. The regularizer weight λ is chosen as $\lambda = \lambda_0 \|Y\|_F^2$, in order to scale with the first term in (P2). Previous works [31] showed that the condition number and spectral norm of the optimal parent transform \hat{W} approach 1 and $1/\sqrt{2}$ respectively, as $\lambda_0 \rightarrow \infty$ in (P2).

Problem (P2) imposes sparsity constraint $\|X_i\|_0 \leq s$ on the sparse code of each signal or image patch. One can also impose an overall sparsity constraint on the entire sparse code matrix X to allow variable sparsity levels (see Section 4.3). Alternatively, a sparsity penalty method can be used, instead of imposing sparsity constraints, which normally leads to efficient algorithms (see Section 4.2). We will demonstrate the use of various sparsity methods in Section 4.

3. FRIST Learning Algorithm and Convergence Analysis

3.1. Learning Algorithm

We propose an efficient algorithm for solving (P2), which alternates between a *sparse coding and clustering* step, and a *transform update* step.

Sparse Coding and Clustering. Given the training matrix Y , and fixed parent transform W , we solve the following Problem (P3) for the sparse codes and clusters,

$$(\text{P3}) \quad \min_{\{C_k\}, \{X_i\}} \sum_{k=1}^K \sum_{i \in C_k} \|W\Phi_k Y_i - X_i\|_2^2 \quad \text{s.t. } \|X_i\|_0 \leq s \quad \forall i, \quad \{C_k\} \in \Gamma$$

The modeling error $\|W\Phi_k Y_i - X_i\|_2^2$ serves as the clustering measure corresponding to signal Y_i , where the best sparse code with FR permutation Φ_k^\dagger is $X_i = H_s(W\Phi_k Y_i)$. Problem (P3) is clearly equivalent to finding the “optimal” FR permutation $\Phi_{\hat{k}_i}$ for each data vector that minimizes this measure by clustering. Clustering of each signal Y_i can thus be decoupled into the following optimization problem,

$$\min_{1 \leq k \leq K} \|W\Phi_k Y_i - H_s(W\Phi_k Y_i)\|_2^2 \quad \forall i \quad (1)$$

where the minimization over k for each Y_i determines the optimal $\Phi_{\hat{k}_i}$, or the cluster $C_{\hat{k}_i}$ to which Y_i belongs. The corresponding optimal sparse code for Y_i in (P3) is thus $\hat{X}_i = H_s(W\Phi_{\hat{k}_i} Y_i)$. Given the sparse code \hat{X}_i , one can also easily recover a least squares estimate of each signal as $\hat{Y}_i = \Phi_{\hat{k}_i}^T W^{-1} \hat{X}_i$. Since the Φ_k 's are permutation matrices, applying and computing Φ_k^T (which is also a permutation matrix) is cheap.

Transform Update Step. We solve for W in (P2) with fixed $\{C_k\}$ and $\{X_i\}$, which leads to the following problem:

$$(P4) \quad \min_W \|W\tilde{Y} - X\|_F^2 + \lambda Q(W)$$

where $\tilde{Y} = [\Phi_{\hat{k}_1} Y_1 \mid \Phi_{\hat{k}_2} Y_2 \mid \dots \mid \Phi_{\hat{k}_N} Y_N]$ contains signals after applying their optimal FR operations, and the columns of X are the corresponding sparse codes X_i 's. Problem (P4) has a simple solution involving a singular value decomposition (SVD), which is similar to the transform update step in SST [35]. We first decompose the positive-definite matrix $\tilde{Y}\tilde{Y}^T + \lambda I_n = UU^T$ (e.g., using Cholesky decomposition). Then, denoting the full singular value decomposition (SVD) of the matrix $U^{-1}\tilde{Y}X^T = S\Sigma V^T$, where $S, \Sigma, V \in \mathbb{R}^{n \times n}$, an optimal transform \hat{W} in (P4) is

$$\hat{W} = 0.5V \left(\Sigma + (\Sigma^2 + 2\lambda I_n)^{\frac{1}{2}} \right) S^T U^{-1} \quad (2)$$

where $(\cdot)^{\frac{1}{2}}$ above denotes the positive definite square root, and I_n is the $n \times n$ identity.

Initialization Insensitivity and Cluster Elimination. Unlike the previously proposed OCTOBOS learning algorithm [41], which requires initialization of the clusters using heuristic methods such as K-means, the FRIST learning algorithm only needs initialization of the parent transform W . In Section 5.1, numerical results demonstrate the fast convergence of the proposed FRIST learning algorithm, which is insensitive to parent transform initialization. In practice, we apply a heuristic cluster elimination strategy in the FRIST learning algorithm, to select the desired K operators. In the first iteration, all possible FR operators Φ_k 's [22, 48] (i.e., all available child transforms W_k 's) are considered for sparse coding and clustering. After each clustering

[†] The FR operator is $\Phi_k = G_q F$, where both G_q and F are permutation matrices. Therefore the composite operator Φ_k is a permutation matrix.

[‡] The sparse code includes the value of \hat{X}_i , as well as the membership index \hat{k}_i which adds just $\log_2 K$ bits to the code storage.

Table 1: Computational cost comparison among SST ($W \in \mathbb{R}^{n \times n}$), OCTOBOS (K clusters, each $W_k \in \mathbb{R}^{n \times n}$), FRIST and KSVD ($D \in \mathbb{R}^{n \times m}$) learning. N is the amount of training data.

	SST.	OCTOBOS	FRIST	KSVD
Cost	$O(n^2N)$	$O(Kn^2N)$	$O(Kn^2N)$	$O(mn^2N)$

step, the learning algorithm eliminates half of the operators with smallest cluster sizes, until the number of selected operators drops to K , which only takes few iterations. For the rest of the iterations, the algorithm only considers the selected K Φ_k 's in the sparse coding and clustering steps.

Computational Cost Analysis. The *sparse coding and clustering* step computes the optimal sparse codes and clusters, with $O(Kn^2N)$ cost. In the transform update step, we compute the closed-form solution for the square parent transform. The cost of the closed-form solution scales as $O(n^2N)$, assuming $N \gg n$, which is cheaper than the sparse coding step. Thus, the overall computational cost per iteration of FRIST learning using the proposed alternating algorithm scales as $O(Kn^2N)$, which is typically lower than the cost per iteration of the overcomplete KSVD learning algorithm, with the number of the dictionary atoms $m = Kn$. We observe that compared to KSVD, the FRIST learning algorithm normally requires fewer iterations to converge. The computational costs per-iteration of SST, OCTOBOS, FRIST, and KSVD learning are summarized in Table 1.

3.2. Convergence Analysis

We analyze the convergence behavior of the proposed FRIST learning algorithm that solves (P2), assuming that every step in the algorithms (such as SVD) is computed exactly.

Notation. Problem (P2) is formulated with a sparsity constraint, which is equivalent to the unconstrained formulation with a sparsity barrier penalty $\phi(X_i)$ (which equals to $+\infty$ when the constraint is violated, and zero otherwise). Thus, the objective function of Problem (P2) can be rewritten as

$$f(W, X, \Lambda) = \sum_{k=1}^K \sum_{i \in C_k} \{ \|W\Phi_k Y_i - X_i\|_2^2 + \phi(X_i) \} + \lambda Q(W) \quad (3)$$

where $\Lambda \in \mathbb{R}^{1 \times N}$ is the vector whose i^{th} element $\Lambda_i \in \{1, \dots, K\}$, which denotes the cluster label k , corresponding to the signal Y_i , $i \in C_k$. We use $\{W^t, X^t, \Lambda^t\}$ to denote the output in each iteration t , generated by the proposed FRIST learning algorithm.

Main Results. As FRIST can be interpreted as structured OCTOBOS, the convergence results of the FRIST learning algorithm take a form similar to that obtained

for the OCTOBOS learning algorithms [41] in recent works. The convergence result for the FRIST learning algorithm, solving (P2), is summarized in the following theorem and corollaries.

Theorem 1 *For each initialization (W^0, X^0, Λ^0) , the following conclusions hold.*

- (i) *The objective $\{f^t\}$ in the FRIST learning algorithm is monotone decreasing, and converges to a finite value, $f^* = f^*(W^0, X^0, \Lambda^0)$.*
- (ii) *The iterate sequence $\{W^t, X^t, \Lambda^t\}$ is bounded, with all of its accumulation points equivalent, i.e., achieving the exactly same value f^* .*
- (iii) *Every accumulation point $\{W, X, \Lambda\}$ of the iterate sequence satisfies the following partial global optimality conditions*

$$(X, \Lambda) \in \arg \min_{\tilde{X}, \tilde{\Lambda}} f(W, \tilde{X}, \tilde{\Lambda}) \quad (4)$$

$$W \in \arg \min_{\tilde{W}} f(\tilde{W}, X, \Lambda) \quad (5)$$

- (iv) *For each accumulation point $\{W, X, \Lambda\}$, there exists $\epsilon = \epsilon(W) > 0$ such that,*

$$f(W + dW, X + \Delta X, \Lambda) \geq f(W, X, \Lambda) = f^* \quad (6)$$

which holds for all $dW \in \mathbb{R}^{n \times n}$ satisfying $\|dW\|_F \leq \epsilon$, and all $\Delta X \in \mathbb{R}^{n \times N}$ satisfying $\|\Delta X\|_\infty < \min_k \min_{i \in C_k} \{\psi_s(W\Phi_k Y_i) : \|W\Phi_k Y_i\|_0 > s\}$. Here, we define the infinity norm of matrix ΔX as $\|\Delta X\|_\infty \triangleq \max_{i,j} |\Delta X_{i,j}|$, and the operator $\psi_s(\cdot)$ to return the s^{th} largest magnitude of a vector.

Conclusion (iv) provides the local optimality condition for each accumulation point w.r.t. (W, X) , where the local perturbation dW in Equation (6) is sufficiently small, and ΔX is specified by a finite region, which is determined by a scalar κ that limits the amplitudes of entries in ΔX (i.e., $\|\Delta X\|_\infty < \kappa$). Here $\kappa = \min_k \kappa_k$, and each $\kappa_k = \min_{i \in C_k} \{\psi_s(W\Phi_k Y_i) : \|W\Phi_k Y_i\|_0 > s\}$ is computed by (i) choosing the vectors with sparsity $> s$ from $\{W\Phi_k Y_i\}$ where $i \in C_k$, (ii) selecting the s^{th} largest of those vectors, and (iii) returning the smallest of those values.

Corollary 1 *For a particular initial (W^0, X^0, Λ^0) , the iterate sequence in FRIST learning algorithm converges to an equivalence class of accumulation points, which are also partial minimizers satisfying (4), (5), and (6).*

Corollary 2 *The iterate sequence $\{W^t, X^t, \Lambda^t\}$ in the FRIST learning algorithm is globally convergent (i.e., it converges from any initialization) to the set of partial minimizers of the non-convex objective $f(W, X, \Lambda)$.*

For reason of space, we only provide an outline of proofs. The conclusion (i) in Theorem 1 is obvious, as the proposed alternating algorithm solves the sub-problem in each step exactly. The proof of Conclusion (ii) follows the same arguments as in the proofs in Lemma 3 and Lemma 5 in [41]. In Conclusion (iii), Condition (4) can be proved using the arguments for Lemma 7 from [41], while Condition (5) can be proved with the arguments for Lemma 6 from [35]. The last conclusion in Theorem 1 can be shown using arguments from the proof of Lemma 9 in [35].

Theorem 1, and Corollaries 1 and 2 establish that with any initialization (W^0, X^0, Λ^0) , the iterate sequence $\{W^t, X^t, \Lambda^t\}$, generated by the FRIST learning algorithm, converges to an equivalence class of partial minimizers of the objective.

4. Applications

Natural, or biomedical images typically contain a variety of directional features and edges, thus the FRIST model is particularly appealing for applications in image processing and inverse problems. In this section, we consider three such applications, namely image denoising, image inpainting, and blind compressed sensing (BCS)-based magnetic resonance imaging (MRI).

4.1. Image Denoising

Image denoising is one of the most fundamental inverse problems in image processing. The goal is to reconstruct a 2D image, which is vectorized as $y \in \mathbb{R}^P$, from its measurement $z = y + h$, corrupted by noise vector h . Various denoising algorithms have been proposed recently, with state-of-the-art performance [46, 7]. Similar to previous dictionary and transform learning based image denoising methods [9, 41], we propose the following patch-based image denoising formulation using FRIST learning,

$$\begin{aligned}
 \text{(P5)} \quad & \min_{W, \{y_i, x_i, C_k\}} \sum_{k=1}^K \sum_{i \in C_k} \{ \|W \Phi_k y_i - x_i\|_2^2 + \tau \|R_i z - y_i\|_2^2 \} + \lambda Q(W) \\
 & \text{s.t. } \|x_i\|_0 \leq s_i \quad \forall i, \quad \{C_k\} \in \Gamma
 \end{aligned}$$

where $R_i \in \mathbb{R}^{n \times P}$ denotes the patch extraction operator, i.e., $R_i z \in \mathbb{R}^n$ represents the i th overlapping patch of the corrupted image z as a vector. We assume N overlapping patches in total. The data fidelity term $\tau \|R_i z - y_i\|_2^2$ between the observed patch $R_i y$ and the noiseless patch y_i uses a weight τ that is set inversely proportional to the given noise level σ [9, 35]. The vector $x_i \in \mathbb{R}^n$ represents the sparse code of y_i in the FRIST domain, with an a priori unknown sparsity level s_i . We follow the previous SST-based and OCTOBOS-based denoising methods [29, 41], and impose the sparsity constraint on each y_i .

We propose a simple iterative denoising algorithm based on (P5). Each iteration involves the following steps: (i) sparse coding and clustering, (ii) sparsity level update, and (iii) transform update. Once the iterations complete, we have a denoised image

reconstruction step. We initialize the $\{y_i\}$ in (P5) using the noisy image patches $\{R_i z\}$. Step (i) is the same as described in Section 3.1. We then update the sparsity levels s_i for all i , similar to the SST learning-based denoising algorithm [29]. With fixed W and clusters $\{C_k\}$, we solve for y_i in (P5) in the least squares sense,

$$y_i = \Phi_k^T \begin{bmatrix} \sqrt{\tau} I \\ W \end{bmatrix}^\dagger \begin{bmatrix} \sqrt{\tau} v_i \\ H_{s_i}(W v_i) \end{bmatrix} = G_1 v_i + G_2 H_{s_i}(W v_i) \quad (7)$$

where G_1 and G_2 are appropriate matrices in the above decomposition, and $v_i \triangleq \Phi_k R_i z$ are the rotated noisy patches, which can be pre-computed in each iteration. We choose the optimal s_i to be the smallest integer that makes the reconstructed y_i satisfy the error condition $\|R_i z - y_i\|_2^2 \leq n C^2 \sigma^2$, where C is a constant parameter [29]. Once step (ii) is completed, we proceed to the transform update based on the method in Section 3.1. The algorithm alternates for a fixed number of iterations, and the denoised image patches $\{y_i\}$ are obtained using (7). Each pixel in the reconstructed patch is projected onto the intensity range (image pixel is typically stored as 8-bit integer, which corresponds to the intensity range $[0, 255]$). The denoised image is reconstructed by averaging the overlapping denoised patches at their respective image locations.

For improved denoising, the algorithm for (P5) is repeated for several passes by replacing z with the most recent denoised image estimate in each pass. The noise level in each such pass is set empirically.

4.2. Image Inpainting

The goal of image inpainting is to recover missing pixels in an image. The given image measurement, with missing pixel intensities set to zero, is denoted as $z = \Xi y + \varepsilon$, where ε is the additive noise on the available pixels, and $\Xi \in \mathbb{R}^{P \times P}$ is a diagonal binary matrix with zeros at locations corresponding to missing pixels. We propose the following patch-based image inpainting formulation using FRIST learning,

$$(P6) \quad \min_{W, \{y_i, x_i, C_k\}} \sum_{k=1}^K \sum_{i \in C_k} \{ \|W \Phi_k y_i - x_i\|_2^2 + \tau^2 \|x_i\|_0 + \gamma \|P_i y_i - z_i\|_2^2 \} + \lambda Q(W)$$

where $z_i = R_i z$ and $y_i = R_i y$. The diagonal binary matrix $P_i \in \mathbb{R}^{n \times n}$ captures the available (non-missing) pixels in z_i . The sparsity penalty $\tau^2 \|x_i\|_0$ is used which leads to an efficient algorithm. The fidelity term $\gamma \|P_i y_i - z_i\|_2^2$ for the i th patch has the coefficient γ that is inversely proportional to the noise standard deviation σ . The parameter τ is proportional to σ , and also increases as more pixels are missing in z .

Our proposed iterative algorithm for solving (P6) involves the following steps: (i) sparse coding and clustering, and (ii) transform update. Once the iterations complete, we have a (iii) patch reconstruction step. The sparse coding problem with a sparsity penalty has closed-form solution [33], and thus Step (i) is equivalent to solving the

following problem,

$$\min_{1 \leq k \leq K} \|W\Phi_k y_i - T_\tau(W\Phi_k y_i)\|_2^2 \quad \forall i \quad (8)$$

where the hard thresholding operator $T_\tau(\cdot)$ is defined as

$$(T_\tau(b))_j = \begin{cases} 0 & , \quad |b_j| < \tau \\ b_j & , \quad |b_j| \geq \tau \end{cases} \quad (9)$$

where the vector $b \in \mathbb{R}^n$, and the subscript j indexes its vector entries. Step (ii) is similar to that in the denoising algorithm in Section 5.4.

Ideal image inpainting without noise. In the ideal case when the noise ε is absent, i.e., $\sigma = 0$, the coefficient of the fidelity term $\gamma \rightarrow \infty$. Thus the fidelity term can be replaced with hard constraints $P_i y_i = z_i \quad \forall i$. In the noiseless reconstruction step, with fixed $\{x_i, C_k\}$ and W , we first reconstruct each image patch y_i by solving the following linearly constrained least squares problem:

$$\min_{y_i} \|W\Phi_{k_i} y_i - x_i\|_2^2 \quad \text{s.t.} \quad P_i y_i = z_i \quad (10)$$

We define $z_i = P_i y_i \triangleq y_i - e_i$, where $e_i = (I_n - P_i)y_i$. Because Φ_k only rearranges pixels, $\Phi_k e_i$ has the support $\Omega_i = \text{supp}(\Phi_k e_i) = \{j \mid (\Phi_k e_i)_j \neq 0\}$, which is complementary to $\text{supp}(\Phi_k z_i)$. Since the constraint leads to the relationship $y_i = z_i + e_i$ with z_i given, we solve the equivalent minimization problem over e_i as follow,

$$\min_{e_i} \|W\Phi_k e_i - (x_i - W\Phi_k z_i)\|_2^2 \quad \text{s.t.} \quad \text{supp}(\Phi_k e_i) = \Omega_i \quad (11)$$

Here, we define W_{Ω_i} to be the submatrix of W formed by columns indexed in Ω_i , and $(\Phi_k e_i)_{\Omega_i}$ to be the vector containing the non-zero entries of $\Phi_k e_i$. Thus, $W\Phi_k e_i = W_{\Omega_i}(\Phi_k e_i)_{\Omega_i}$, and we define $\xi^i \triangleq \Phi_k e_i$. The reconstruction problem is then re-written as the following unconstrained problem,

$$\min_{\xi_{\Omega_i}^i} \|W_{\Omega_i} \xi_{\Omega_i}^i - (x_i - W\Phi_k z_i)\|_2^2 \quad \forall i \quad (12)$$

The above least squares problem has a simple solution given as $\hat{\xi}_{\Omega_i}^i = W_{\Omega_i}^\dagger (x_i - W\Phi_k z_i)$. Accordingly, we can calculate $\hat{e}_i = \Phi_k^T \hat{\xi}_{\Omega_i}^i$, and thus the reconstructed patches $\hat{y}_i = \hat{e}_i + z_i$.

Robust image inpainting. We now extend to noisy z , and propose the robust inpainting algorithm. This is useful because real image measurements are inevitably corrupted with noise [16]. The robust reconstruction step for each patch is to solve the following problem,

$$\min_{y_i} \|W\Phi_{k_i} y_i - x_i\|_2^2 + \gamma \|P_i y_i - z_i\|_2^2 \quad (13)$$

Let $\tilde{z}_i \triangleq \Phi_{k_i} z_i$, $u_i \triangleq \Phi_{k_i} y_i$, and $\tilde{P}_i \triangleq \Phi_{k_i} P_i \Phi_{k_i}^T$, where Φ_{k_i} is permutation matrix. The rotated solution \hat{u}_i in optimization problem (13) is equivalent to

$$\hat{u}_i = \arg \min_{u_i} \|W u_i - x_i\|_2^2 + \gamma \|\tilde{P}_i u_i - \tilde{z}_i\|_2^2 \quad (14)$$

which has a least square solution $\hat{u}_i = (W^T W + \gamma \tilde{P}_i)^{-1} (W^T x_i + \gamma \tilde{P}_i \tilde{z}_i)$. As the matrix inversion $(W^T W + \gamma \tilde{P}_i)^{-1}$ is expensive with a cost of $O(n^3)$ for each patch reconstruction, we apply Woodbury Matrix Identity and rewrite the solution to (14) as

$$\hat{u}_i = [B - B_{\Upsilon_i}^T (\frac{1}{\gamma} I_{q^i} + \Psi_i)^{-1} B_{\Upsilon_i}] (W^T x_i + \gamma \tilde{P}_i \tilde{z}_i) \quad (15)$$

where $B \triangleq (W^T W)^{-1}$ can be pre-computed, and the support of \tilde{z}_i is denoted as $\Upsilon_i \triangleq \text{supp}(\tilde{z}_i)$. The scalar $q^i = |\Upsilon_i|$ counts the number of available pixels in z_i . Here, B_{Υ_i} is the submatrix of B formed by B_{Υ_i} -indexed rows, while Ψ_i is the submatrix of B_{Υ_i} formed by B_{Υ_i} -indexed columns. Thus, the matrix inversion $(\frac{1}{\gamma} I_{q^i} + \Psi_i)^{-1}$ has cost of $O((q^i)^3)$, compared to computing $(B + \gamma \tilde{P}_i)^{-1}$ with cost of $O(n^3)$ for the reconstruction of each patch. For an inpainting problem with most pixels missing ($q^i \ll n$), this represents significant saving. In the supplementary, with few pixels missing ($q^i \approx n$), a similar procedure can be used with $I - P_i$ replacing P_i . Once \hat{u}_i is computed, the patch in (13) is recovered as $\hat{g}_i = \Phi_{k_i}^T \hat{u}_i$.

Similar to Section 4.1, each pixel in the reconstructed patch is projected onto the intensity range (e.g., $[0, 255]$ for image pixel stored using 8-bit integer). Eventually, we output the inpainted image by averaging the reconstructed patches at their respective image locations. We perform multiple passes in the inpainting algorithm for (P6) for improved inpainting. In each pass, we initialize $\{y_i\}$ using patches extracted from the most recent inpainted image. By doing so, we indirectly reinforce the dependency between overlapping patches in each pass.

4.3. BCS-based MRI

Compressed Sensing (CS) enables accurate MRI reconstruction from far fewer measurements than required by Nyquist sampling [38, 32, 15]. However, CS-based MRI suffers from various artifacts at high undersampling rate, when using non-adaptive analytical transforms [38]. Recent works [32] proposed BCS-based MRI methods using adaptively learned sparsifying transforms, and generated superior reconstruction results. Furthermore, MRI image patches typically contain various oriented features [48], which have recently been shown to be well sparsifiable by directional wavelets [26]. Compared to directional analytical transforms, FRIST can adapt to the MRI data by unsupervised learning, while clustering the image patches simultaneously based on their geometric orientations, which leads to more accurate sparse modeling of MRI image.

Based on the previous TL-MRI work [32], we propose a BCS-based MRI imaging scheme using adaptively learned FRIST, dubbed FRIST-MRI. For computational efficiency, we restrict the parent W to be unitary transform, instead of well-conditioned transform. The FRIST-MRI problem with sparsity constraint is formulated as

$$\begin{aligned} \text{(P7)} \quad & \min_{W, y, \{x_i, C_k\}} \mu \|F_u y - z\|_2^2 + \sum_{k=1}^K \sum_{i \in C_k} \|W \Phi_k R_i y - x_i\|_2^2 \\ & \text{s.t. } W^H W = I, \|X\|_0 \leq s, \|y\|_2 \leq L, \{C_k\} \in \Gamma \end{aligned}$$

where $W^H W = I$ is the unitary constraint, $y \in \mathbb{C}^P$ is the MRI image to be reconstructed, and $z \in \mathbb{C}^M$ denotes the measurements with the sensing matrix $F_u \in \mathbb{C}^{M \times P}$, which is the undersampled Fourier encoding matrix. Here $M \ll P$, as Problem (P7) is aimed to reconstruct MRI image y from highly undersampled measurements z . The constraint $\|y\|_2 \leq L$ represents prior knowledge of the signal range, with some $L > 0$. The sparsity term $\|X\|_0$ counts the number of non-zeros in the entire sparse matrix $X \in \mathbb{C}^{n \times P}$, whose columns are the sparse codes $\{x_i\}$. This sparsity constraint enables variable sparsity levels for individual patches [32].

We use the block coordinate descent-type approach [32] to solve the FRIST-MRI reconstruction problem (P7). The proposed algorithm alternates between (i) sparse coding and clustering, (ii) parent transform update, and (iii) MRI image reconstruction. We initialize the FRIST-MRI algorithm with the zero-filled Fourier reconstruction $F_u^H z$ for y . Step (i) solves Problem (P7) for $\{x_i, C_k\}$ with fixed W and y as

$$\min_{\{x_i, C_k\}} \sum_{k=1}^K \sum_{i \in C_k} \|W \Phi_k R_i y - x_i\|_2^2 \quad s.t. \quad \|X\|_0 \leq s, \quad \{C_k\} \in \Gamma \quad (16)$$

The exact solution to Problem (16) requires calculating the sparsification error for each possible clustering. The cost scales as $O(Pn^2K^P)$, which is computationally infeasible. Instead, we provide an approximate solution, which we observed to work well in our experiment. In this method, we first compute the total sparsification error SE_k , associated with each Φ_k , by solving the following problem,

$$SE_k = \sum_{i=1}^P SE_k^i \triangleq \min_{\{\beta_i^k\}} \sum_{i=1}^P \|W \Phi_k R_i y - \beta_i^k\|_2^2 \quad s.t. \quad \|B^k\|_0 \leq s \quad (17)$$

where the columns of B^k are $\{\beta_i^k\}$. The optimal clusters $\{C_k\}$ in (16) are approximately computed, by assigning $i \in C_{\hat{k}}$ where $\hat{k} = \arg \min_k SE_k^i$. Once the clusters are computed,

the sparse codes \hat{X} in (16) (for fixed clusters) are found by thresholding the matrix $[W \Phi_{\hat{k}_1} R_1 y \mid \dots \mid W \Phi_{\hat{k}_P} R_P y]$ and retaining the s largest magnitude elements [32].

Step (ii) updates the parent transform W with unitary constraint. The solution, which is similar to previous work [35], is exact with fixed $\{x_i, C_k\}$ and y . We first calculate the full SVD $AX^H = \tilde{S} \tilde{\Sigma} \tilde{V}^H$, where the columns of A are $\{\Phi_k R_i y\}$. The optimal unitary parent transform is then $\hat{W} = \tilde{V} \tilde{S}^H$.

Step (iii) solves for y with fixed W and $\{x_i, C_k\}$ as

$$\min_y \sum_{k=1}^K \sum_{i \in C_k} \|W \Phi_k R_i y - x_i\|_2^2 + \mu \|F_u y - z\|_2^2 \quad s.t. \quad \|y\|_2 \leq L \quad (18)$$

As Problem (18) is a least squares problem with ℓ_2 constraint, it can be solved exactly using the Lagrange multiplier method [12], which is equivalently to solving

$$\min_y \sum_{k=1}^K \sum_{i \in C_k} \|W \Phi_k R_i y - x_i\|_2^2 + \mu \|F_u y - z\|_2^2 + \rho (\|y\|_2^2 - L) \quad (19)$$

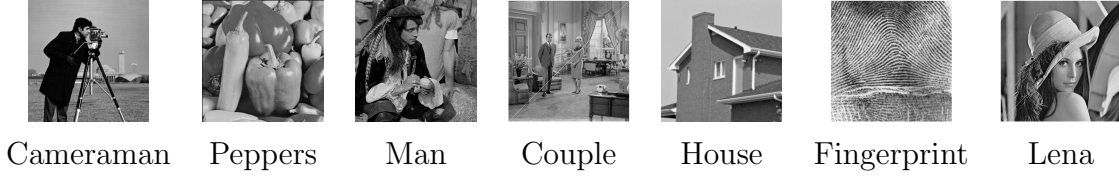


Figure 1: Testing images used in the image denoising and image inpainting the experiments.

where $\rho \geq 0$ is the optimally chosen Lagrange multiplier. Similar to the simplification that was proposed in previous TL-MRI work [32], the normal equation of Problem (19) can be simplified as

$$(FEF^H + \mu FF_u^H F_u F^H + \rho I)Fy = F \sum_{k=1}^K \sum_{i \in C_k} R_i^H \Phi_k^H W^H x_i + \mu FF_u^H z \quad (20)$$

where $E \triangleq \sum_{k=1}^K \sum_{i \in C_k} R_i^H \Phi_k^H W^H W \Phi_k R_i = \sum_{i=1}^P R_i^H R_i$. As FEF^H , $\mu FF_u^H F_u F^H$, and ρI are all diagonal matrices, the matrix which pre-multiplies Fx is diagonal and invertible. Using unitary W constraint leads to efficient update of y here. In particular, the matrix E is not easily diagonalizable when W is not unitary. The signal y (and the optimal ρ) is then reconstructed efficiently using the method proposed by the TL-MRI work [32].

5. Experiments

We present numerical convergence result of the FRIST learning algorithm, image segmentation, as well as some preliminary results demonstrating the promise of FRIST learning in applications including image sparse representation, denoising, robust inpainting, and MRI reconstruction. We work with 8×8 non-overlapping patches for the study of convergence and sparse representation, 8×8 overlapping patches for image segmentation, denoising, and robust inpainting, and 6×6 overlapping patches (including wrap-around patches) for MRI experiments. Figure.1 lists the testing images that are used in image denoising and inpainting experiments.

5.1. Empirical convergence results

We first illustrate the convergence behavior of FRIST learning. We randomly extract 10^4 non-overlapping patches from the 44 images in the USC-SIPI database [1] (the color images are converted to gray-scale images), and learn a FRIST, with a 64×64 parent transform W , from the randomly selected patches using fixed sparsity level $s = 10$. We set $K = 2$, and $\lambda_0 = 3.1 \times 10^{-3}$ for visualization simplicity. In the experiment, we initialize the learning algorithm with different square 64×64 parent transform W 's, including (i) Karhunen-Loève Transform (KLT), (ii) 2D DCT, (iii) random matrix with

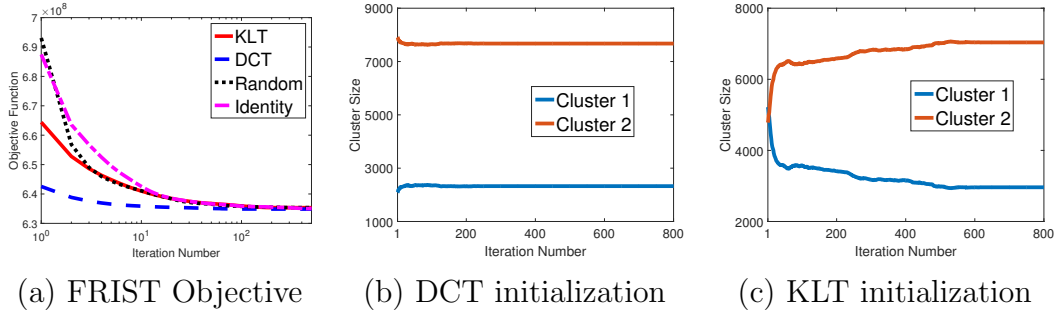


Figure 2: Convergence of FRIST objective and cluster size with various parent transform initializations.

i.i.d. Gaussian entries (zero mean and standard deviation 0.2), and (iv) the identity matrix. Figure 2(a) illustrates the convergence behavior of the objective functions over iterations, with different parent W initializations. The final values of the objective are identical for all the initializations. Figure 2(b) and Figure 2(c) show the cluster size changes over iterations for 2D DCT and KLT initializations. The final values of the cluster sizes are similar (although, not necessarily identical) for various initializations.

The numerical results demonstrate that our FRIST learning algorithm is reasonably robust, or insensitive to initialization. Good initialization for the parent transform W , such as DCT, leads to faster convergence of learning. Thus, we initialize parent transform W using 2D DCT, in the rest of the experiments.

5.2. Image Segmentation and Clustering Behavior

The FRIST learning algorithm is capable of clustering image patches according to their orientations. In this subsection, we illustrate the FRIST clustering behavior by an image segmentation experiment. We consider the images *Wave* (512×512) and *Field* (512×512) shown in Fig. 3(a) and Fig. 4(a) as inputs. Both images contains directional textures, and we aim to cluster the pixels of the images into one of the four classes, which represent different orientations. For each input image, we convert it into gray-scale, extract the overlapping mean-subtracted patches, and learn a FRIST while clustering the patches using algorithm in Section 3.1. As overlapping patches are used, each pixel in the image may belong to several classes. We cluster a pixel into a particular class by majority voting, between the patches that contain it.

We set $s = 10$, and $K = 4$ in the clustering experiments. Figure 3 and Fig. 4 illustrate the segmentation results of images *Wave* and *Field*. Figure 3(b) and Fig. 4(b) illustrate the pixel membership with four different colors (blue, red, green, and black, for class 1 to 4 respectively). Figures 3(c)-(f) and Figs. 4(c)-(f) each visualize the image pixels clustered into a specific class in gray-scale, and the pixels that are not clustered into that class are shown in black. The parent transform W and its child transforms W_k 's in the learned FRIST for *Wave* are visualized in Fig. 5 with rows of each W_k .

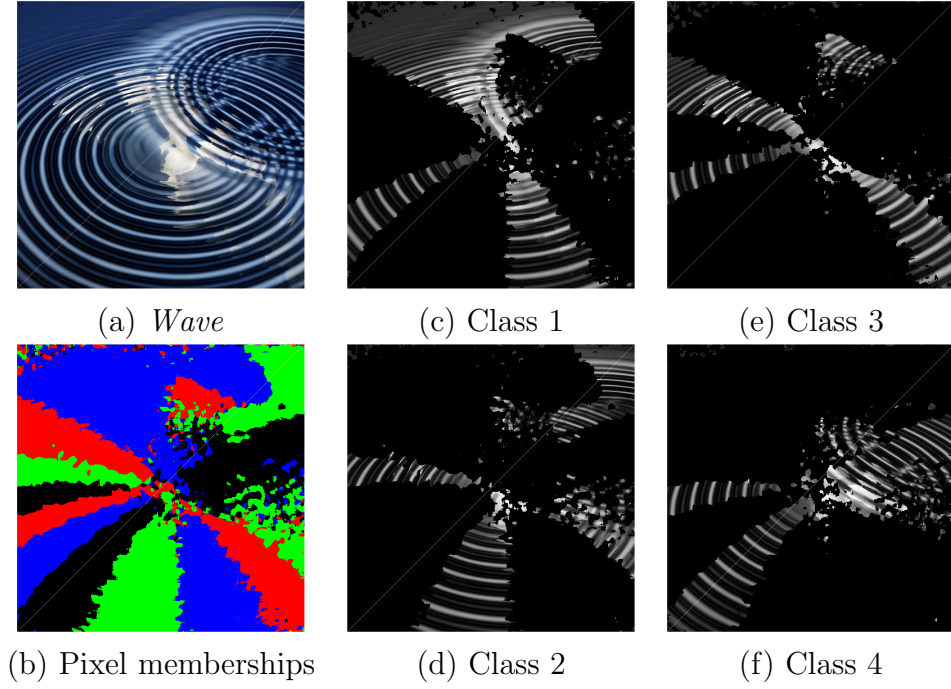


Figure 3: Image segmentation result of *Wave* (512×512) using FRIST learning on the gray-scale version of the image. The Four different colors present pixels that are belong to the four classes. Pixels that are clustered into a specific class are shown in the gray-scale, while pixels that are not clustered into that class, are shown in black for (c)-(f).

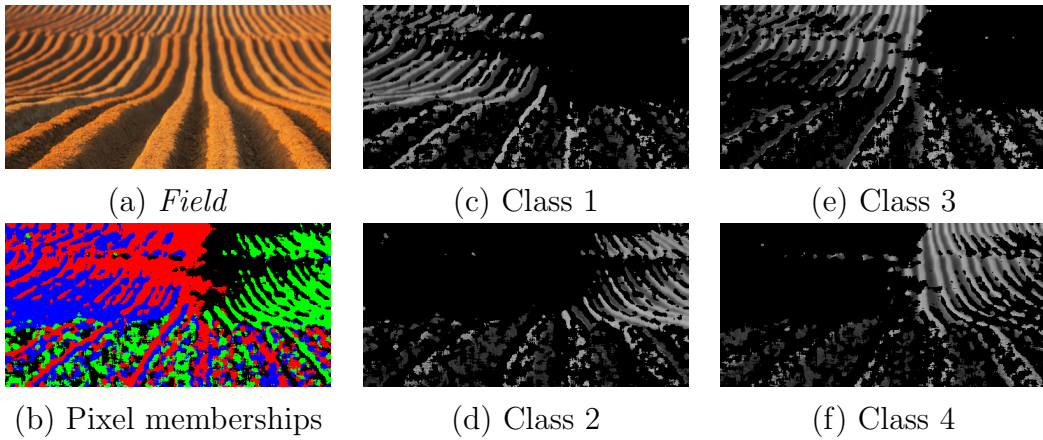


Figure 4: Image segmentation result of *Field* (256×512) using FRIST learning on the gray-scale version of the image. The Four different colors present pixels that are belong to the four classes. Pixels that are clustered into a specific class are shown in the gray-scale, while pixels that are not clustered into that class, are shown in black for (c)-(f).

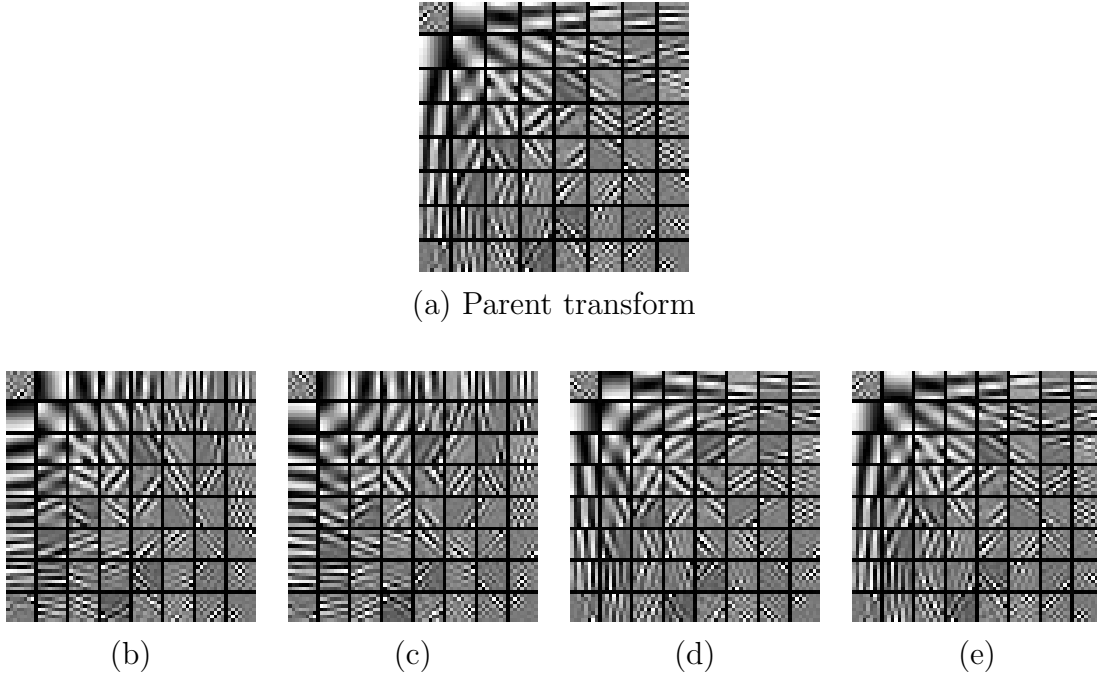


Figure 5: Visualization of the learned (a) parent transform, and (b)-(e) child transforms in FRIST for image *Wave*. The rows of each child transform are displayed as patches.

displayed as patches. We observe that each child transform contains distinct directional features that were adaptively learned, in order to sparsify edges with various orientations better. The parent W turns out to be identical to the child transform shown in Fig. 4(d), meaning that the corresponding FR operator is identity matrix. We also observed reasonable clustering / image segmentation results with other images.

The preliminary image segmentation results demonstrate some potential for the FRIST scheme for directional classification. More importantly, we wish to illustrate why FRIST can provide improvement over SST or OCTOBOS in various inverse problems. As natural images usually contain directional textures, FRIST is capable of grouping those patches with similar orientations, and thus provides better sparsification in each cluster by learning directional child transforms.

5.3. Image Sparse Representation

Most of the popular image compression methods make use of analytical sparsifying transforms. In particular, the commonly used JPEG uses the 2D DCT to sparsify image patches. Data-driven adaptation of dictionaries using the K-SVD scheme has also been shown to be beneficial for image compression, compared to fixed analytical transforms [5]. In this section, we do not attempt to demonstrate a complete image compression scheme providing state-of-the-art performance, but rather show that the proposed FRIST learning scheme provides improved image sparse representation, which can be potentially applied in an adaptive image compression framework.

Table 2: PSNR values for reconstruction of images from sparse representation obtained using the 2D DCT, SST, OCTOBOS, square and overcomplete K-SVD, and our proposed FRIST method. The first row of the table provides average PSNR values computed over the 44 images from USC-SIPI database. The best PSNR values are marked in bold.

Methods Model Size	2D DCT	SST	OCTOBOS	K-SVD		FRIST
		64×64	128×64	64×64	64×128	64×64
<i>USC-SIPI</i>	34.36	34.20	33.62	34.11	35.08	35.14
<i>Cameraman</i>	29.49	29.43	29.03	29.09	30.16	30.63
<i>House</i>	36.89	36.36	35.38	36.31	37.41	37.71

We learn a FRIST, with a 64×64 parent transform W , from the 10^4 randomly selected patches (from USC-SIPI images) used in Section 5.1. We set $K = 32$, $s = 10$ and $\lambda_0 = 3.1 \times 10^{-3}$. To compare with other popular adaptive sparse models, we also train a 64×64 SST [35], a 128×64 OCTOBOS [41], as well as a 64×64 square (synthesis) dictionary and a 64×128 overcomplete dictionary using KSVD [9] using the same training patches.

With the learned models, we represent each image from the USC-SIPI database compactly, by storing its sparse representation, including (i) non-zeros in the sparse codes of the 8×8 non-overlapping patches, (ii) locations of the non-zeros (plus the cluster membership if necessary) which needs only a small overhead cost, and (iii) the adaptive sparse model. For each method, the patch sparsity (or equivalently, the number of (i) non-zeros in each patch) is set consistently as $s = 10$. The adaptive SST, square KSVD, and FRIST methods store only a 64×64 square matrix, whereas the overcomplete KSVD and OCTOBOS methods store a 128×64 matrix.

The images are then reconstructed from their sparse representations in a least squares sense, and the reconstruction quality for each image is evaluated using Peak-Signal-to-Noise Ratio (PSNR), expressed in decibels (dB). We use the average of the PSNR values over all 44 images as the indicator of the quality of compression of the USC-SIPI database. Additionally, we apply the learned sparse models to represent some standard images that are not included in the USC-SIPI database.

Table 2 lists the sparse representation reconstruction results. We observe that our learned FRIST provides the best reconstruction quality compared to other adaptive sparse models or analytical 2D DCT, for both the images in the database and the external images. Compared to unstructured overcomplete models such as KSVD and OCTOBOS, our proposed FRIST provides promising image sparse representation quality, while requiring fewer bits to represent the model itself by maintaining lower model richness. We only need to store a small parent transform, and the cluster membership indices for the FRIST model. Additionally, dictionary learning based

Table 3: PSNR values (in dB) for denoising with 64×64 adaptive FRIST along with the corresponding PSNR values for denoising using the 64×64 2D DCT, the 64×64 SST, the 64×256 overcomplete K-SVD, the 256×64 OCTOBOS, and BM3D. The best PSNR values are marked in bold.

Image	σ	Noisy PSNR	DCT	SST	K-SVD	OCTOBOS	BM3D	FRIST
<i>Peppers</i> (256×256)	5	34.14	37.70	37.95	37.78	38.09	38.09	38.16
	10	28.10	34.00	34.37	34.24	34.57	34.66	34.68
	15	24.58	31.83	32.14	32.18	32.43	32.69	32.54
<i>Cameraman</i> (256×256)	5	34.12	37.77	38.01	37.82	38.16	38.21	38.16
	10	28.14	33.63	33.90	33.72	34.13	34.15	34.16
	15	24.61	31.33	31.65	31.51	31.95	31.91	31.97
<i>Man</i> (768×768)	5	34.15	36.59	36.64	36.47	36.73	36.76	36.82
	10	28.13	32.86	32.95	32.71	32.98	33.18	33.06
	15	24.63	30.88	30.96	30.78	31.07	31.32	31.10
<i>Lena</i> (512×512)	5	34.16	38.52	38.62	38.61	38.72	38.70	38.72
	10	28.12	35.36	35.52	35.49	35.64	35.88	35.67
	15	24.64	33.51	33.68	33.72	33.91	34.26	33.93
<i>Couple</i> (512×512)	5	34.16	37.25	37.32	37.29	37.40	37.48	37.43
	10	28.11	33.48	33.60	33.50	33.73	34.01	33.78
	15	24.59	31.35	31.47	31.44	31.71	32.08	31.71

representation requires synthesis sparse coding, which is more expensive compared to the cheap and exact sparse coding in transform model based methods [31]. The investigation of a complete image compression scheme using FRIST, with quantitative experiments and analysis is left to future work.

5.4. Image Denoising

We present denoising results using our FRIST-based framework in Section 4.1. We simulate i.i.d. Gaussian noise at 3 different noise levels ($\sigma = 5, 10, 15$) for five standard images. Denoising results obtained by our proposed algorithm in Section 4.1 are compared, with those obtained by the adaptive overcomplete K-SVD denoising scheme [9], adaptive SST denoising scheme [35], adaptive OCTOBOS denoising scheme [41], and BM3D [7], which is a state-of-the-art image denoising method. We also compare to the denoising result using the SST method, but with fixed 2D DCT.

We set $K = 64$, $n = 64$, $C = 1.04$ for the FRIST denoising method. For the adaptive SST and OCTOBOS denoising methods, we follow the same parameter settings proposed in the previous works [35, 41]. The same parameter settings for the SST method is used for DCT-based denoising algorithm. A corresponding 64×256 synthesis

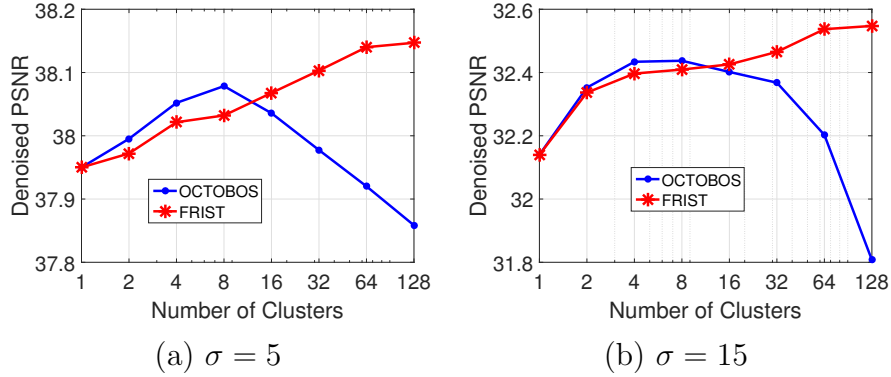


Figure 6: Denoising PSNR for *Peppers* as a function of the number of cluster K .

dictionary is used in the synthesis K-SVD denoising method, to match with the same model richness as the 256×64 OCTOBOS. For the K-SVD, and BM3D methods, we use the publicly available implementations in this experiment.

Table 3 lists the denoised PSNR results. The proposed FRIST scheme constantly provides better PSNRs compared to other fixed or adaptive sparse modeling methods including DCT, SST, K-SVD, and OCTOBOS for all testing bases. Figure 6 plots the denoising PSNRs for *Peppers* as a function of the number of child transforms (i.e., the number of clusters) K for $\sigma = 5$ and $\sigma = 15$. In both cases, the denoising PSNRs of the OCTOBOS and FRIST schemes increase with K initially. Beyond an optimal value of K , the OCTOBOS denoising scheme suffers from overfitting which causes the denoising PSNR to decrease [41]. As seen by comparing Fig. 6(a) and Fig. 6(b), this effect is more pronounced the higher the noise level. Instead, the FRIST based denoising scheme, which has constant number of degrees of freedom while K increases, provides monotonically increasing denoising PSNR. Compared to BM3D, FRIST can also provide comparable denoising PSNRs. We expect the denoising PSNRs for FRIST to improve further with optimal parameter tuning.

5.5. Image Inpainting

We present preliminary results for our adaptive FRIST-based inpainting framework (based on (P6)). We randomly remove 80% and 90% of the pixels of the entire image, and simulate i.i.d. additive Gaussian noise for the sampled pixels with $\sigma = 0, 5, 10$, and 15. We set $K = 64$, $n = 64$, and apply the proposed FRIST inpainting algorithm to reconstruct the image from the highly corrupted and noisy measurements. We additionally replace the adaptive FRIST in the proposed inpainting algorithm with fixed 2D DCT, adaptive SST [35], and OCTOBOS [41], and evaluate the inpainting performance using the corresponding models for comparison. The image inpainting results obtained by the FRIST based methods are also compared with those obtained by the cubic interpolation [45, 39] and patch smoothing [28] methods. We used the Matlab function “griddata” to implement the cubic interpolation, and use the publicly available

Table 4: PSNR values for image inpainting, averaged over six images, using adaptive FRIST based method, along with the corresponding values obtained using cubic interpolation (Cubic), patch smooth ordering (Smooth), patch-based DCT, adaptive SST, and adaptive OCTOBOS based methods, with various fractions of available pixels and noise levels. The best PSNR value in each row is marked in bold.

Available pixels	σ	Corrupted PSNR	Cubic	Smooth	DCT	SST	OCTOBOS	FRIST
20%	0	6.40	26.56	28.87	29.23	29.25	29.27	29.33
	5	6.39	6.40	28.64	29.21	29.24	29.26	29.31
	10	6.37	6.37	27.07	28.13	28.73	28.99	29.16
	15	6.33	6.33	25.52	26.94	28.07	28.44	28.67
10%	0	5.89	24.02	25.77	26.14	26.16	26.14	26.20
	5	5.87	5.88	25.46	25.59	25.85	25.96	26.08
	10	5.86	5.86	24.67	25.02	24.98	25.27	25.46
	15	5.82	5.82	23.73	24.19	24.47	24.68	24.88

implementations of the patch smoothing method. For the DCT, SST, OCTOBOS and FRIST based methods, we initialize the image patches using the Cubic Interpolation method in noiseless cases, and using the Patch Smoothing method in noisy cases.

Table 4 lists the image inpainting PSNR results, averaged over the images shown in Fig. 1, with various fractions of sampled pixels and noise levels. The proposed FRIST inpainting scheme provides better PSNRs compared to all other inpainting methods based on interpolation, transform-domain sparsity, and spatial similarity. Figure 7 provides an illustration of the inpainting results, with regional zoom-in for visual comparison. We observe that the cubic interpolation produces blur in various locations. The Cubic Interpolation method is extremely sensitive to noise, whereas the FRIST based method is the most robust. Compared to the competitors, FRIST provides larger inpainting PSNR improvement as the noise level and the amount of missing pixels increase, by using a highly constrained adaptive overcomplete sparse model.

5.6. MRI Reconstruction

We present preliminary MRI reconstruction results using the proposed FRIST-MRI algorithm. The three complex-valued images and the corresponding k-spaces sampling masks (with various sampling schemes and undersampling rate), with which we work in this section, are shown in Fig. 8, Fig. 9 (a), and Fig. 9 (b) §. We set $K = 32$, the sparsity level $s = 0.05 \times nP$, and other parameters similar to those used in [32] for reconstructing

§ The testing image data in this section were used and included in previous works [32, 48] with the data sources.

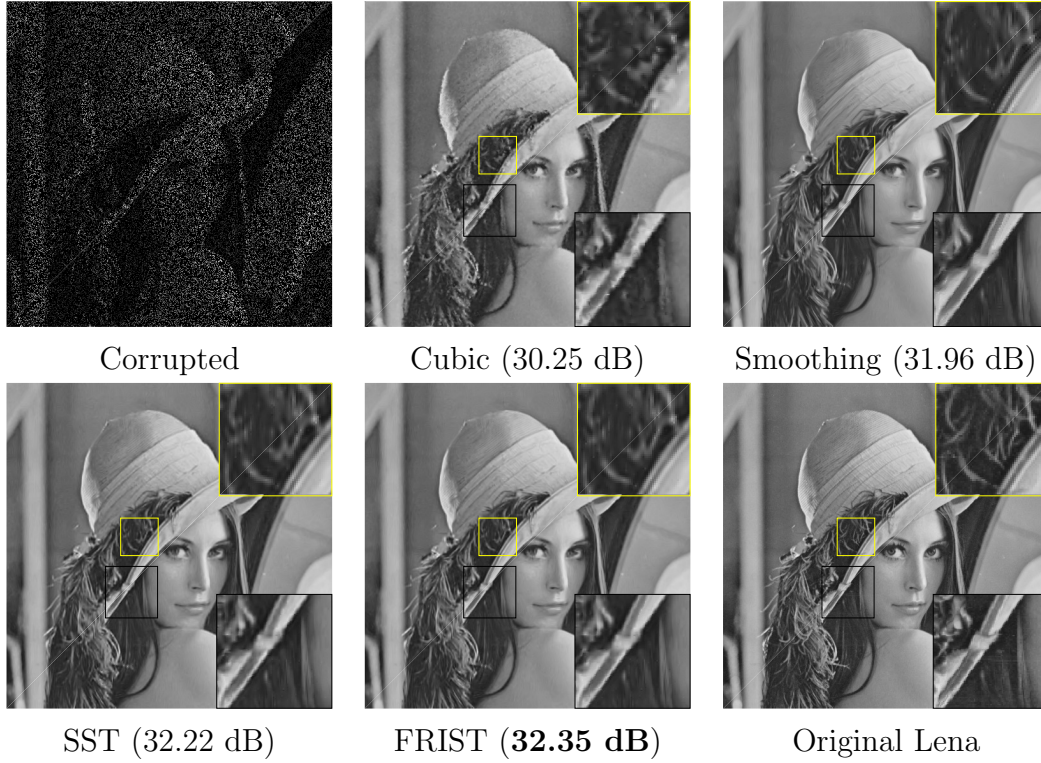


Figure 7: Illustration of image inpainting results for *Lena*, with regional zoom-in comparisons.

Image **1** and Image **2**. We used a higher sparsity level $s < 0.085 \times nN$ for reconstructing Image **3** which contains more smooth regions. To speed up convergence, lower sparsity level is used in the initial iterations [32]. We compare our FRIST-MRI reconstruction results to those obtained using conventional or popular methods, including Zero-filling, Sparse MRI [15], DL-MRI [34], PBDWS [20], PANO [27], and TL-MRI [32]. In order to make fair comparison to the TL-MRI method, we tune the sparsity parameter for constructing Image **3** ¶. The comparison of the reconstruction PSNRs are listed in Table 5.

First, our proposed FRIST-MRI algorithm provides significant improvements over the Zero-filling reconstruction (the initialization of the algorithm) with 6.4 dB higher PSNR, as well as over the Sparse MRI reconstruction with 4.2 dB higher PSNR, averaged over all testing data. Comparing to recently proposed popular MRI reconstruction methods, the FRIST-MRI algorithm demonstrates reasonably better performance for each testing case. We obtained an average PSNR improvement of 0.8 dB, 0.5 dB, and 0.3 dB for FRIST-MRI over the non-local patch similarity-based PANO method, the partially adaptive PBDWS method, and the adaptive dictionary-based DL-MRI method. We observed that the MRI methods with adaptively learned regularizer usually provide

¶ We observed improved reconstruction PSNR compared to the result obtained using the sparsity level described in [32].

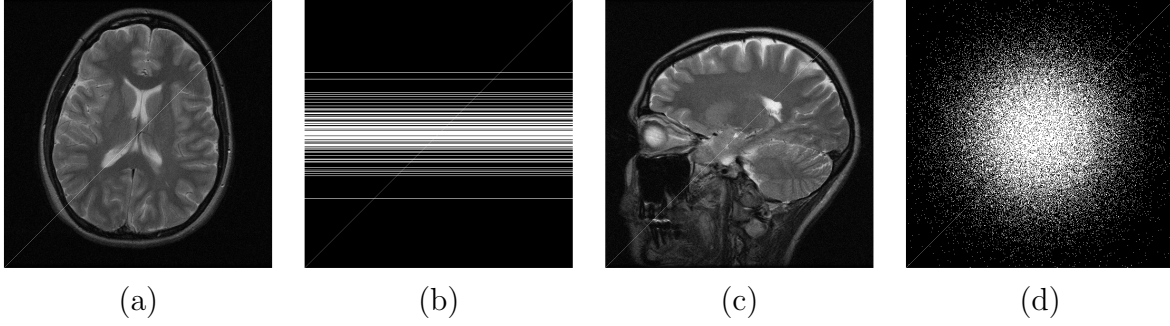


Figure 8: Testing MRI images and their k-space sampling masks: (a) Image **1**; (b) k-space sampling mask (Cartesian with $7\times$ undersampling) for Image **1**; (c) Image **2**; (d) k-space sampling mask (2D random with $5\times$ undersampling) for Image **2**.

Table 5: Comparison of the PSNRs, corresponding to the Zero-filling, Sparse MRI [15], DL-MRI [34], PBDWS [20], PANO [27], TL-MRI [32], and the proposed FRIST-MRI reconstructions for various images, sampling schemes, and undersampling factors. The best PSNR for each MRI image is marked in bold.

Image	Sampling Scheme	Under-sampl.	Zero-filling	Sparse MRI	DL-MRI	PBDWS	PANO	TL-MRI	FRIST-MRI
1	Cartesian	$7\times$	27.9	28.6	30.9	31.1	31.1	31.2	31.4
2	2D Random	$5\times$	26.9	27.9	30.5	30.3	30.4	30.6	30.7
3	Cartesian	$2.5\times$	24.9	29.9	36.6	35.8	34.8	36.3	36.7

much better reconstruction quality, compared to those using fixed models.

The proposed FRIST-MRI reconstruction quality is somewhat better than TL-MRI, with an 0.2 dB PSNR improvement on average. As we followed a reconstruction framework and parameters similar to those used by the TL-MRI method, the quality improvement using FRIST-MRI is solely because the learned FRIST can serve as a better regularizer for MRI image reconstruction, compared to the single square transform. We also observed that FRIST-MRI used lower average sparsity level than TL-MRI for each testing case, and thus it provides better sparse representations for MRI images. Figure 9 visualizes the reconstruction quality comparison between FRIST-MRI and TL-MRI. The magnitude of FRIST-MRI reconstruction error clearly shows fewer artifacts, especially along the edges of the circles, compared to TL-MRI.

6. Conclusion

In this paper, we presented a novel framework for rotation and flipping invariant sparsifying transform learning. In particular, we proposed a method for learning a

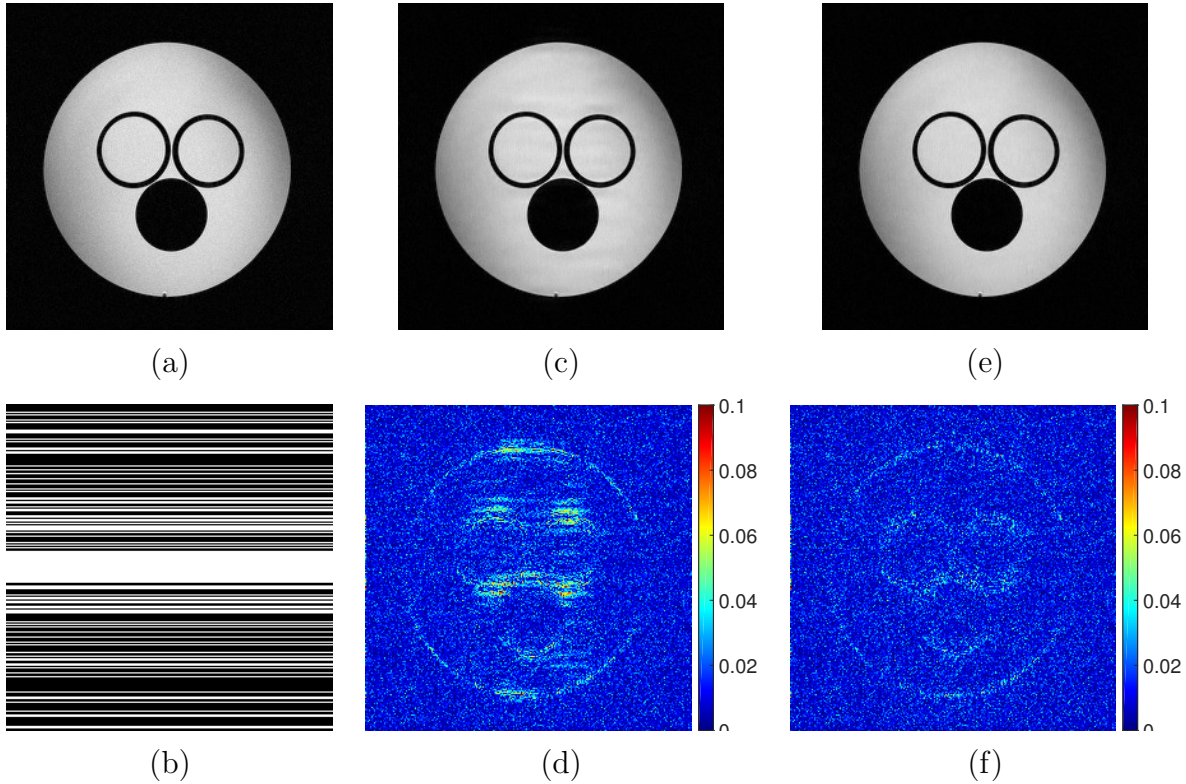


Figure 9: Visualization of reconstruction result of Image **3** using Cartesian and $2.5\times$ undersampling: (a) Image **3**; (b) sampling mask in k-space; (c) TL-MRI reconstruction (36.3 dB); (d) magnitude of TL-MRI reconstruction error; (e) FRIST-MRI reconstruction (36.7 dB); (f) magnitude of FRIST-MRI reconstruction error.

structured union-of-transforms model, dubbed FRIST. The collection of transforms in FRIST are related to a generating parent transform, by flipping and rotation operations. Our algorithm for FRIST learning is highly efficient, and involves closed-form solution with convergence guarantee. We have demonstrated the ability of FRIST learning in extracting directional features in images. FRIST is insensitive to initialization, and performs better than several prior methods in applications, including sparse image representation, image denoising, image inpainting, and MRI reconstruction (blind compressed sensing).

References

- [1] The USC-SIPI Image Database. [Online: <http://sipi.usc.edu/database/database.php?volume=misc>; accessed July-2014].
- [2] M. Aharon and M. Elad. Sparse and redundant modeling of image content using an image-signature-dictionary. *SIAM Journal on Imaging Sciences*, 1(3):228–247, 2008.
- [3] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.

- [4] A. M. Bruckstein, D. L. Donoho, and M. Elad. From sparse solutions of systems of equations to sparse modeling of signals and images. *SIAM Review*, 51(1):34–81, 2009.
- [5] O. Bryt and M. Elad. Compression of facial images using the k-svd algorithm. *Journal of Visual Communication and Image Representation*, 19(4):270–282, 2008.
- [6] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998.
- [7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3D transform-domain collaborative filtering. *IEEE Trans. on Image Processing*, 16(8):2080–2095, 2007.
- [8] G. Davis, S. Mallat, and M. Avellaneda. Adaptive greedy approximations. *Journal of Constructive Approximation*, 13(1):57–98, 1997.
- [9] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Process.*, 15(12):3736–3745, 2006.
- [10] M. Elad, P. Milanfar, and R. Rubinstein. Analysis versus synthesis in signal priors. *Inverse Problems*, 23(3):947–968, 2007.
- [11] K. Engan, S.O. Aase, and J.H. Hakon-Husoy. Method of optimal directions for frame design. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 2443–2446, 1999.
- [12] Sivan Gleichman and Yonina C Eldar. Blind compressed sensing. *IEEE Transactions on Information Theory*, 57(10):6958–6975, 2011.
- [13] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 506 – 513, 2004.
- [14] D. G. Lowe. Object recognition from local scale-invariant features. In *IEEE International Conference on Computer vision (ICCV)*, volume 2, pages 1150–1157, 1999.
- [15] Michael Lustig, David Donoho, and John M Pauly. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.
- [16] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Trans. on Image Processing*, 17(1):53–69, 2008.
- [17] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, 11:19–60, 2010.
- [18] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, 1999.
- [19] S. G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.
- [20] Bende Ning, Xiaobo Qu, Di Guo, Changwei Hu, and Zhong Chen. Magnetic resonance image reconstruction using trained geometric directions in 2d redundant wavelets domain and non-convex optimization. *Magnetic resonance imaging*, 31(9):1611–1622, 2013.
- [21] Y. Pati, R. Rezaifar, and P. Krishnaprasad. Orthogonal matching pursuit : recursive function approximation with applications to wavelet decomposition. In *Asilomar Conf. on Signals, Systems and Comput.*, pages 40–44 vol.1, 1993.
- [22] E. L. Pennec and S. Mallat. Bandelet image approximation and compression. *Multiscale Modeling & Simulation*, 4(3):992–1039, 2005.
- [23] Luke Pfister and Y. Bresler. Adaptive sparsifying transforms for iterative tomographic reconstruction. In *International Conference on Image Formation in X-Ray Computed Tomography*, 2014.
- [24] Luke Pfister and Y. Bresler. Learning sparsifying filter banks. In *Proc. SPIE Wavelets & Sparsity XVI*, 2015.
- [25] W. K. Pratt, J. Kane, and H. C. Andrews. Hadamard transform image coding. *Proc. IEEE*, 57(1):58–68, 1969.
- [26] X. Qu, D. Guo, B. Ning, Y. Hou, Y. Lin, S. Cai, and Z. Chen. Undersampled MRI reconstruction with patch-based directional wavelets. *Magnetic resonance imaging*, 30(7):964–977, 2012.
- [27] Xiaobo Qu, Yingkun Hou, Fan Lam, Di Guo, Jianhui Zhong, and Zhong Chen. Magnetic resonance

- image reconstruction from undersampled measurements using a patch-based nonlocal operator. *Medical image analysis*, 18(6):843–856, 2014.
- [28] I. Ram, M. Elad, and I. Cohen. Image processing using smooth ordering of its patches. *IEEE Transactions on Image Processing*, 22(7):2764–2774, 2013.
 - [29] S. Ravishankar and Y. Bresler. Learning doubly sparse transforms for images. *IEEE Trans. Image Process.*, 22(12):4598–4612, 2013.
 - [30] S. Ravishankar and Y. Bresler. Learning overcomplete sparsifying transforms for signal processing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3088–3092, 2013.
 - [31] S. Ravishankar and Y. Bresler. Learning sparsifying transforms. *IEEE Trans. Signal Process.*, 61(5):1072–1086, 2013.
 - [32] S. Ravishankar and Y. Bresler. Efficient blind compressed sensing using sparsifying transforms with convergence guarantees and application to magnetic resonance imaging. *SIAM Journal on Imaging Sciences*, 8(4):2519–2557, 2015.
 - [33] S. Ravishankar, B. Wen, and Y. Bresler. Online sparsifying transform learning - part i: Algorithms. *IEEE Journal of Selected Topics in Signal Process.*, 9(4):625–636, 2015.
 - [34] Saiprasad Ravishankar and Yoram Bresler. MR image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE Transactions on Medical Imaging*, 30(5):1028–1041, 2011.
 - [35] Saiprasad Ravishankar and Yoram Bresler. ℓ_0 sparsifying transform learning with efficient optimal updates and convergence guarantees. *IEEE Transactions on Signal Processing*, 63(9):2389–2404, 2014.
 - [36] Ron Rubinstein, Alfred M. Bruckstein, and Michael Elad. Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057, 2010.
 - [37] K. Skretting and K. Engan. Recursive least squares dictionary learning algorithm. *IEEE Transactions on Signal Processing*, 58(4):2121–2130, 2010.
 - [38] Joshua Trzasko and Armando Manduca. Highly undersampled magnetic resonance image reconstruction via homotopic-minimization. *IEEE Transactions on Medical imaging*, 28(1):106–121, 2009.
 - [39] D. Watson. *Contouring: a guide to the analysis and display of spatial data*. Elsevier, 2013.
 - [40] B. Wen, S. Ravishankar, and Y. Bresler. Learning overcomplete sparsifying transforms with block cosparsity. In *IEEE International Conference on Image Processing (ICIP)*, 2014.
 - [41] B. Wen, S. Ravishankar, and Y. Bresler. Structured overcomplete sparsifying transform learning with convergence guarantees and applications. *Int. J. Computer Vision*, 114(2):137–167, 2015.
 - [42] H. Wersing, J. Eggert, and E. Körner. Sparse coding with invariance constraints. In *Artificial Neural Networks and Neural Information Processing ICANN/ICONIP*, pages 385–392. 2003.
 - [43] M. Woodbury. Inverting modified matrices. *Memorandum report*, 42:106, 1950.
 - [44] M. Yaghoobi, T. Blumensath, and M. Davies. Dictionary learning for sparse approximations with the majorization method. *IEEE Transaction on Signal Processing*, 57(6):2178–2191, 2009.
 - [45] T. Yang. *Finite element structural analysis*, volume 2. Prentice Hall, 1986.
 - [46] G. Yu and G. Sapiro. DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line*, 1, 2011.
 - [47] L. Zelnik-Manor, K. Rosenblum, and Y. Eldar. Dictionary optimization for block-sparse representations. *IEEE Transactions on Signal Processing*, 60(5):2386–2395, 2012.
 - [48] Z. Zhan, J. Cai, D. Guo, Y. Liu, Z. Chen, and X. Qu. Fast multi-class dictionaries learning with geometrical directions in MRI reconstruction. *arXiv preprint arXiv:1503.02945*, 2015.